2009-03

Privacy for mobile networks via network virtualization

Glidden, Todd P.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/4867

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

PRIVACY FOR MOBILE NETWORKS VIA NETWORK VIRTUALIZATION

by

Todd P. Glidden

March 2009

Thesis Advisor:	Gurminder Singh
Co-Advisor:	John Gibson

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Privacy for Mobile Networks via Network Virtualization			**5. FUNDING NUMBERS**
6. AUTHOR(S) Glidden, Todd P.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Today mobile devices have become powerful and ubiquitous. The conveniences afforded by these devices do not come without a cost, however. The use of mobile devices and mobile networks poses a significant risk to privacy. Four privacy requirements for mobile networks are identified: content privacy, identity privacy, location privacy, and authentication. This work focuses on content privacy. Two threats to content privacy are identified: the casual observer and the attacker. This work seeks to provide content privacy protection against the identified threats in mobile networks used by first responders. TwiddleNet, a mobile network designed for the data dissemination requirements of first responders, was used as a platform for implementation.

A network virtualization technique was used in order to provide content privacy protection. This allows TwiddleNet users to share content on a per-group basis among virtual networks of user groups. It was found that this virtualization technique successfully provided content privacy protection from the threat of a casual observer, but not from an attacker. Providing adequate protections from the attacker threat requires more sophisticated measures and is left to future work.

14. SUBJECT TERMS Privacy, Mobile networks, First responders, Mobile file sharing, Data dissemination			**15. NUMBER OF PAGES** 65
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

PRIVACY FOR MOBILE NETWORKS VIA NETWORK VIRTUALIZATION

Todd P. Glidden
Lieutenant Commander, United States Navy
B.E., S.U.N.Y Maritime College, 1997

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2009

Author: Todd P. Glidden

Approved by: Dr. Gurminder Singh
Thesis Advisor

John Gibson
Co-Advisor

Dr. Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Today mobile devices have become powerful and ubiquitous. The conveniences afforded by these devices do not come without a cost, however. The use of mobile devices and mobile networks poses a significant risk to privacy. Four privacy requirements for mobile networks are identified: content privacy, identity privacy, location privacy, and authentication. This work focuses on content privacy. Two threats to content privacy are identified: the casual observer and the attacker. This work seeks to provide content privacy protection against the identified threats in mobile networks used by first responders. TwiddleNet, a mobile network designed for the data dissemination requirements of first responders, was used as a platform for implementation.

A network virtualization technique was used in order to provide content privacy protection. This allows TwiddleNet users to share content on a per-group basis among virtual networks of user groups. It was found that this virtualization technique successfully provided content privacy protection from the threat of a casual observer, but not from an attacker. Providing adequate protections from the attacker threat requires more sophisticated measures and is left to future work.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First and foremost, I would like to thank my wife Joy, and my family, for their support during my long hours of study and writing, not only during preparation of my thesis, but over the duration of my two-year tenure here at NPS.

Additionally, I would like to thank LT Dirk Ableiter, German Navy, for his inspiration and support during the early stages of my work with TwiddleNet. Also, many thanks go to LCDR Lillian Abuan for her help and enthusiasm during our trips to Camp Roberts for TwiddleNet testing. Furthermore, many thanks go to my thesis advisors, Dr. Gurminder Singh and John Gibson, for their guidance in support of this thesis work and for their excellent instruction in the many classes that I took from them.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Over the last several years, mobile devices have become increasingly capable and ubiquitous. The term *mobile device* in the context of this work includes cell phones, smart phones, and network-enabled Personal Digital Assistants (PDAs). The popularity of these devices has led to some striking numbers in terms of market penetration. According to research firm Informa [1], in November 2007, worldwide mobile phone subscriptions reached 3.3 billion – equivalent to half the global population. Of course this number is an aggregate value for all the world's nations. Some countries have higher penetration rates than others. Eurostat, an organization that tracks statistics for European Commission nations, reported that there are nearly 95 mobile phones for every 100 Europeans [2]. Penetration is as high as 158% in Lithuania [2] meaning some people own multiple devices.

In addition to being extremely popular, mobile devices are more powerful than ever. The processing power of today's mobile devices is greater than that of the desktop PCs used in the 1990s [3]. The content capture and communication capabilities of modern mobile devices have opened the door for a wide variety of new services. In addition to voice, text, and e-mail services that have always been popular, it is now possible for mobile devices to support streaming multimedia and video conferencing. Large companies, like Microsoft and Intel, have recognized the value of the growing mobile technology market and are working vigorously to capitalize upon the demand. The importance of mobile technology to such companies was seen at the January 2008 Consumer Electronics Show where Microsoft chairman Bill Gates remarked, "The trend now is to have information wherever you want [4]." This trend echoes the needs of the military as well as first responders for "the right information, at the right place, at the right time [5]."

The power and ubiquity of mobile devices does not come without a cost, however. The use of mobile devices and networks comes with significant privacy challenges. This thesis explores an implementation of privacy protection for mobile

networks used by first responders via network virtualization. Herein network virtualization is defined as methods used to create a logically and/or physically separated set of resources running on top of a single physical network. TwiddleNet, a wireless system comprised of mobile devices, will be used as a platform for proof-of-concept and implementation.

A. DEFINITION OF PRIVACY

Privacy can mean many things depending on the context of the situation. Calcutt broadly defines privacy as "the right of the individual to be protected against intrusion into his personal life or affairs … by direct physical means or by publication of information [6]." As noted in [7], privacy can be broken down into four separate concepts: information privacy, bodily privacy, privacy of communications, and territorial privacy. Information privacy refers to controls placed on the collection and handling of personal data, such as credit information, and medical and government records. Bodily privacy deals with the protection of people's physical selves against invasive procedures, such as genetic tests, undue drug testing, and cavity searches. The security and privacy of mail, telephones, e-mail and other forms of communication is covered by privacy of communications. Territorial privacy involves the setting of limits on intrusion into the domestic and other environments such as the workplace or public space.

This thesis will concentrate on information privacy and privacy of communications, as the other concepts are not related to the context of this work. From here on these concepts will more generally be referred to as "content privacy."

B. PRIVACY REQUIREMENTS FOR MOBILE NETWORKS

Users of mobile networks may require privacy so as to prevent an attacker from learning information, such as where they live, where they work, whom they communicate with, and what they say. These privacy issues have been a topic of a great deal of study. Privacy requirements most often identified in literature include content privacy, identity privacy, location privacy, and authentication.

Content privacy was previously defined in the discussion of information privacy and privacy of communications. At the most basic level, content privacy can be thought of as confidentiality. Users require that their information is not accessible to unauthorized parties either during transmission across the network or when stored in a record system.

Identity privacy deals with protecting a user's name or other identifiers that could be used to uniquely identify him [8]. Users may want to communicate anonymously or pseudonymously without revealing their actual identity. They may want to reveal their identity only to the other communicating party while remaining anonymous to any intermediaries or eavesdroppers, or they may want to remain anonymous with respect to the communication network itself [8], [9].

Location privacy typically refers to privacy of the user's point of attachment to the network, i.e., their network location or address. The aim of location privacy is to prevent an attacker or observer from linking the two locations to the same user when that user changes points of attachment to the network [10]. Privacy of geographic location can be a concern, too, as it is possible to ascertain the rough location of a transmitter by triangulation or signal analysis; however protection against these methods is beyond the scope of this work.

Authentication refers to a process by which a user verifies that a communicating party is in fact who they claim to be. This is required to prevent an adversary from impersonating a legitimate user and thereby gaining access to privileged information. Ideally, mutual authentication is in place, where a user authenticates himself to the network and the network authenticates itself to the user.

C. PRIVACY REQUIREMENTS FOR FIRST RESPONDERS

1. General Requirements

First responders may come from many disparate organizations and agencies. It is not uncommon to have individuals from fire, police or other law enforcement, medical, military, federal, state, and local government agencies, as well as non-government

agencies, responding to an event. Members from all these groups may require access to the mobile network in order to do their jobs. Each group may have different requirements in terms of the level of privacy needed. Therefore, as a general requirement, each group of first responders needs to be able to keep their information separate from other groups as the situation dictates. A group also should be able to share information with all other groups if desired.

2. Specific Requirements

a. Medical Information

As previously mentioned, TwiddleNet will be the platform used for implementation of this thesis work. A typical use case scenario for TwiddleNet involves emergency medical personnel using the system to gather triage information about victims of a natural disaster or other mass casualty incident. Medical information of this type is subject to requirements delineated in the Health Insurance Portability and Accountability Act (HIPAA) [11], which establishes standards for the security and privacy of patient health information. HIPAA restricts the use and disclosure of patient health information. Therefore, it is desirable to restrict the access to this information to medical personnel only.

b. Privacy Act Information

Information about U.S. citizens or permanent residents collected by first responders working for the Department of Defense (DoD) or other federal government agencies is subject to the provisions of the Privacy Act of 1974 [12]. Such information includes education and medical information, financial transactions, criminal or employment history, and any identifiers assigned to an individual such as a Social Security Number [13]. The purpose of the Privacy Act is to regulate the collection, maintenance, use, and dissemination of personal information by federal agencies. In

keeping with this purpose, any personal information collected by first responders covered by the act should be handled in such a way as to prevent inadvertent disclosure to persons without a need to know.

c. Military Information

First responder teams often include military personnel. Military members typically will have requirements for private communication of sensitive information, such as that pertaining to force protection. In this case, it is desirable to restrict the access of this information to military and perhaps law enforcement personnel.

D. THREAT TO PRIVACY IN MOBILE NETWORKS

This work will consider two general categories of threats to privacy in mobile networks: eavesdropping and surveillance, and the threat posed by the use of unique identifiers.

1. Eavesdropping and Surveillance

Wireless networks are inherently insecure due to the nature of the medium. In a wireless shared medium, assuming all stations are using the same protocols, any station can receive all traffic from other stations that are within range and can transmit to any station within range. Thus, any traffic that is sent unencrypted can be easily intercepted. This represents a threat not only to the content privacy of a user's communications, but also to identity and location privacy, as intercepted traffic may be analyzed to ascertain certain unique identifiers (discussed in the next section) that can lead to a compromise in privacy.

Two types of parties may be seen as posing a threat by eavesdropping or observation: casual observers and attackers. A casual observer herein is defined as another user of the mobile network who may have access to, or receive information sent by a user, but did not actively seek out that information. For example, in the current TwiddleNet implementation all active users of the system receive notification of new content that a user has made available regardless of the user's desire for privacy control.

Addressing this threat is the primary focus of this work. The implementation work done for this thesis allows TwiddleNet users to have more control over the recipients of their notifications. For instance, if a medical responder wishes only to alert other medical users, due to concerns over patient privacy, he can do so, preventing other users, such as fire or police responders, from having access to the information.

The second party seen as posing a threat via eavesdropping is an attacker. An attacker is considered to be an individual who actively seeks out information that he would not otherwise have access to. For instance an attacker may use a “packet sniffer” or protocol analyzer to glean information from the wireless medium. This threat is normally addressed through the use of encryption and authentication. In Wireless Local Area Networks (WLANs) based on the IEEE 802.11 standard, three cryptographic protocols are commonly used: Wired Equivalent Privacy (WEP), the Temporal Key Integrity Protocol (TKIP), and the Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) [14]. WEP and TKIP are based on the Rivest Cipher 4 (RC4) algorithm, and CCMP is based on the Advanced Encryption Standard (AES). In Wireless Wide Area Networks (WWANs), like the Global System for Mobile Communications (GSM) cellular networks, the encryption algorithms used are dependent on the particular service provider [15].

It should be noted that there is a great deal of concern among privacy groups surrounding the threat posed by surveillance of mobile networks (indeed any electronic communications) on the part of law enforcement and intelligence agencies [16], [17]. This is not considered as a threat in this work, as law enforcement agencies, after obtaining a warrant, are authorized to conduct such surveillance and compliance by network operators with this type of surveillance is required by law [18], [19], [20], [21].

2. Unique Identifiers

As mentioned above, encryption is commonly employed to ensure content privacy in mobile networks. A user’s privacy may still be at risk however, due to unique identifiers used in network protocols that may lead to a compromise in identity or

location privacy. Even with encryption in place such identifiers may reveal the user's identity and activities to anyone within transmission range.

An overview of some protection mechanisms aimed at addressing this threat is presented in Chapter II. Implementation of protection measures for unique identifiers is beyond the scope of this work however.

E. OBJECTIVE

The objective of this thesis is to implement privacy measures in mobile networks for first responders using network virtualization. The following research questions will be addressed: 1) Is network virtualization an appropriate choice for implementing privacy protections? 2) How well does the implementation address the stated privacy requirements?

F. SCOPE

The scope of this thesis is limited to providing content privacy protection for mobile networks. TwiddleNet will be used as the platform for implementation. The scope of work is limited to those changes to the code and architecture of the existing TwiddleNet system necessary to accomplish the stated objective. Work will be limited to the applications layer as the use of techniques that require modified kernel modules and network card drivers are not possible on the devices available to the student.

G. ORGANIZATION

The rest of this document is organized as follows. Chapter II covers background information on the TwiddleNet system and related work in the area of privacy protections for mobile networks. Chapter III outlines the design of the new version of TwiddleNet while Chapter IV details implementation aspects of the privacy protection measures. Chapter V concludes and discusses future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. TWIDDLENET OVERVIEW

TwiddleNet is a distributed system of mobile devices. The system harnesses the power of today's mobile devices to create a network of mobile personal servers. The content capture and networking capabilities of the devices allow TwiddleNet users to capture and publish information in real time, while at the same time retaining ownership control of published content. In this way, content is made available that is otherwise not readily accessible to anyone but the device owner.

Designed to be run on handheld devices, TwiddleNet is most useful for first-responder networking and information-sharing tasks that require immediate content capture and dissemination [22]. The system is well suited to first responder applications due to the fact that it runs on lightweight devices, can be set up quickly, and supports real-time information exchange.

The following is a brief overview of the major components that make up TwiddleNet and how they work together in a typical information-sharing scenario.

1. Client

A TwiddleNet Client consists of the Client software running on a mobile device. The current implementation runs on HP iPAQ hw6945 smartphones with the Windows Mobile 5 operating system, utilizing the .NET 2.0 Compact Framework. The Client has three primary functions: 1) to create metadata for new content and notify the Portal of its availability, 2) to provide an interface for a user to discover and download new content, and 3) to serve content to other Clients.

2. Portal

The TwiddleNet Portal consists of the Portal software running on a standard PC running any Windows operating system capable of supporting the .NET 2.0 Framework.

In the current implementation, the Portal is typically run on an OQO ultra mobile PC, which is more portable than a laptop or desktop computer [23].

The Portal's primary function is to act as a gateway to the network of mobile devices. It acts as a central repository for metadata describing shared content. Using the Client software, a user can search for specific content among the metadata residing on the Portal. The Portal also houses information on all TwiddleNet users and keeps track of the IP address currently assigned to each user's device. Additionally, the Portal can cache content, temporarily taking on content serving duties, thus easing the burden on resource-strapped clients.

3. Command Post

The Command Post is a program that performs some of the functions of a TwiddleNet Client. It is meant to run on a Windows laptop or desktop PC. The Command Post is capable of receiving TwiddleNet alerts and automatically retrieves the content associated with each alert it receives. It then builds web pages displaying that content. The web server software that hosts the content is collocated on the machine running the Command Post software.

The Command Post is envisioned to be used at a command center or headquarters. It is intended to serve as a situational awareness tool providing real-time information to the commander of an operation in order to facilitate timely decision making.

4. TwiddleNet Operation

When a user has content to share, he places the content in a designated directory in the device's file system. This may be done manually by the user or automatically by the camera software on the user's device, for example. The TwiddleNet Client software monitors this directory and automatically generates metadata for the content according to the user's preferences. This metadata is then sent to the Portal, serving as a notification that new content is available (Figure 1, Step 1). The metadata is in XML format and includes descriptive items (or "tags"), such as the username of the creator/publisher, file name, file size, etc., as well as user defined tags. A full description of the metadata

generation process is given in [24]. See also [25] for details on a specialized user-defined tagging scheme for medical personnel engaged in triage work.

Once the Portal receives metadata for new content, it alerts other Clients via another XML-based message containing the metadata, notifying them that new content is available (Figure 1, Step 2). The Client software displays a notification to the user upon receipt of such an alert. The user can then choose to download the content (Figure 1, Step 3). The download will be via an HTTP GET method, normally directly from the publisher of the content, unless the content is cached on the Portal. As previously mentioned, content may be temporarily cached on the Portal due to power or bandwidth limitations on the part of the publisher's device. In the case that content is cached, the Portal will retrieve the content from the publishing device using the same mechanism. See [25] for a detailed description of the caching process.

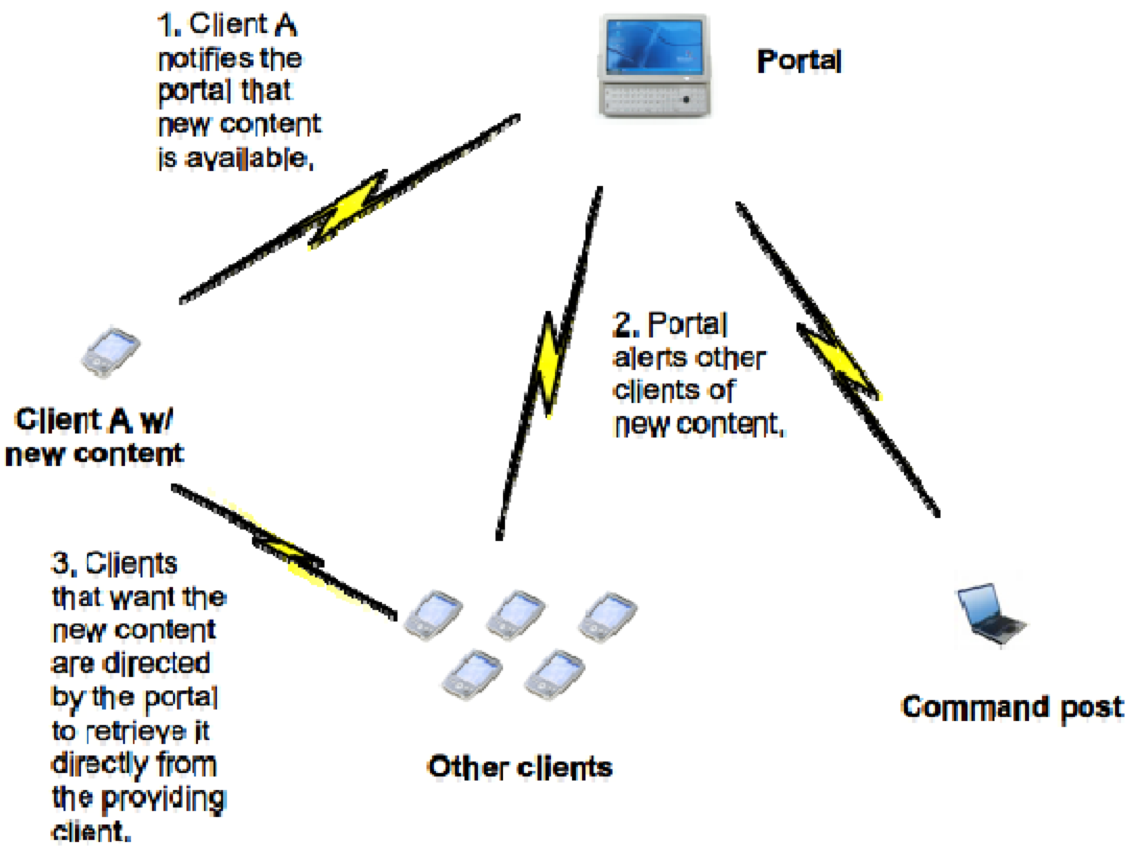


Figure 1. TwiddleNet System Operation

B. RELATED WORK IN PRIVACY PROTECTION FOR MOBILE NETWORKS

This section outlines some common techniques found in the literature for providing privacy protection in mobile networks.

1. Anonymity

Anonymity can be used to provide identity protection for users. It should be noted that identity privacy is closely linked with location privacy: If an attacker cannot correlate an identifier with a particular user as that user moves around in the network, then the attacker cannot track that user's location.

Numerous examples of anonymity being employed in mobile networks can be found in the literature. Aura and Zugenmaier mention the use of proxies to provide anonymity [8]. When a proxy (or any network address translation device) is used, traffic from multiple users behind the device appears to be originated at one address. In this way, a user can "hide in the crowd" among the group of users behind the proxy. An attacker cannot discern one user's traffic from another's based solely on identifiers such as IP or MAC address since all traffic is coming from the address of the proxy. It should be noted that this technique alone cannot prevent an attacker from identifying a user through traffic analysis or indirectly through the context of the communication if the actual content of the communication is accessible.

Digital mixes can also provide anonymity. Askwith, Merabti, Shi, and Whiteley propose the use of a digital mix network in the Global System for Mobile Communications (GSM) [9]. A digital mix enables two parties to communicate without unauthorized parties being able to determine either the message content or the source and destinations of the messages. Thus, a user can communicate anonymously through a digital mix network.

The solution presented in [10] makes use of the Host Identity Protocol (HIP) [26]. HIP allows anonymous identifiers to be created by end users. Users can communicate anonymously to other HIP enabled nodes using these identifiers.

2. Pseudonyms

Pseudonyms are often used to protect identity privacy, and thereby, location privacy, in mobile networks. In Wireless Local Area Networks (WLANs), the pseudonym is normally in the form of a temporary identifier, such as an IP or MAC address. In the implementation discussed in [27], temporary disposable MAC addresses are used. Each time a client creates a new association with an access point it randomly generates a new valid MAC address to use during that session. A similar idea is presented in [28] where a client is given IP and MAC addresses by the access point from a pool of valid addresses each time a new association is created. The idea here is that by frequently changing these unique identifiers, an attacker would not be able to link different communications to a specific user or track that user as he changes locations within the network.

Pseudonyms are also used in Wireless WWANs. In GSM a pseudonym called the Temporary Mobile Subscriber Identity (TMSI) is used to protect the International Mobile Subscriber Identity (IMSI) on the radio path [15]. In a similar manner to that described above, the TMSI can be changed for every call setup.

3. Encryption

Encryption is commonly used to provide content privacy in wireless networks. For instance, the Wired Equivalent Privacy (WEP) and WiFi Protected Access (WPA) encryption protocols are heavily used in 802.11 based WLANs. The keys used by encryption protocols can also be used in support of authentication.

Encryption can also provide identity privacy by hiding identifiers in the application layer. Identifiers like e-mail user names would be hidden in encrypted traffic preventing an attacker from associating the traffic with a particular user [28].

Furthermore, if transport layer identifiers, like port numbers, are hidden, encryption can mask the type of communication (e-mail, web browsing, etc.) being conducted. This can make it more difficult for an attacker to identify a particular user based on knowledge of past activity.

4. Virtualization

Virtualization has been used as a technique to protect user privacy in IP based networks. While most work in this area does not address mobile networks specifically, some could be applied to WLANs at least, if not WWANs as well, provided the mobile devices involved are capable of running the software required. These software requirements typically include running modified kernel modules and/or modified network drivers or virtual machine monitors (VMMs), virtual network interfaces, and virtual switching or routing software.

The approach used in [29] involves protocol stack virtualization where each application being run by a user will have its own virtual network stack. This partitioning scheme addresses linkability, preventing a privacy violation in one application from affecting others. Previously mentioned techniques for protecting identity and location privacy such as the use of temporary addresses can be implemented in each virtual stack on a per-application basis rather than forcing one protocol stack to meet the needs of various different applications.

Cabuk, Dalton, Ramasamy, and Schunter use existing technologies such as Ethernet encapsulation, virtual LAN tagging, and virtual private networks to implement virtual network enclaves [30]. The framework developed by the authors allows for communication between enclaves to be governed by an input security model. While this work doesn't specifically address privacy issues, in theory a system such as this could be used to provide separation between groups of users in a mobile network in order to provide privacy protection.

III. DESIGN

A. OVERVIEW

Since the TwiddleNet system operates at the application layer, the strategy for this implementation focuses on the application layer. The design for this work is such that privacy protection can be achieved by modifying the TwiddleNet software in order to change the system's behavior. The system was modified so that users can be partitioned into groups and information can be shared with only the groups desired by the sharing user, thus providing privacy protection. An application layer implementation is preferable to other strategies mentioned in Chapter II, which require changes at lower levels, because all the application program code is available for inspection and modification. Furthermore, programming at the application layer in a high-level language, such as that used to create the TwiddleNet software, most closely matches the abilities of the author; the author is familiar with application development, but has little experience with operating system or device driver development.

The design of this implementation makes use of a network virtualization approach that allows TwiddleNet users to be partitioned into logical groups that can be combined to form virtual networks in arbitrary combinations. Consider the following scenario as an example. Suppose firefighters, medical personnel, and police are using TwiddleNet in response to a mass-casualty incident. These users can be placed into separate groups according to their role along with, for instance, an additional group for a command center at a hospital. The medical personnel wish to keep patient info they collect private allowing only other medical personnel and the command center to have access to it. The medics will configure their devices to share information only with their own group and the command center group. This effectively creates a separate virtual network, inclusive of only the medical and command center users, on top of the physical TwiddleNet network. Users in the firefighter and police groups, which are outside of this virtual network, will be unaware of any information passed by medical personnel.

The following figures illustrate the user-partitioning effect. Previous implementations of TwiddleNet did not allow for any user groups, and therefore, all users where essentially in a single group as depicted in Figure 2. As Figure 3 shows, the new implementation allows users to be organized into multiple groups while remaining connected to the same physical network. A user in one group can select other groups to share content with, dynamically creating virtual networks of groups as needed.

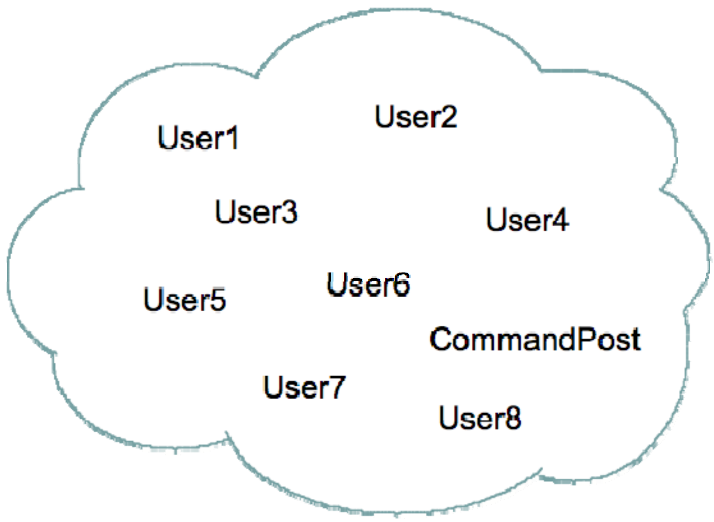


Figure 2. In previous implementations all users were in the same group

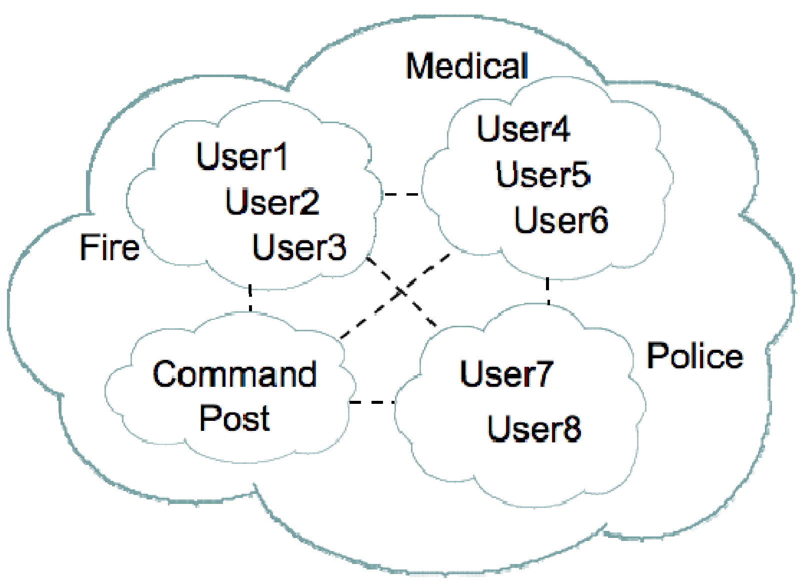


Figure 3. This implementation allows for grouping of users

B. DESIGN ASPECTS

No changes to the physical components of the TwiddleNet architecture were necessary for this design. The new implementation of the TwiddleNet system still consists of the Portal, a number of Client devices, and the Command Post. The difference in this implementation is in system behavior. The design aspects and modifications made to the major TwiddleNet components in order to implement user groups are addressed here.

1. Client

A fundamental aspect of this implementation is that a TwiddleNet user can share information on a per-group basis. For this to be possible of course, the Client needs to be aware of what groups are available. To achieve this, the user sign-in process was modified to allow the Portal to send group information to the Client upon sign-in. These modifications are detailed in Chapter IV. This group information is used to build part of the Client's graphical user interface (GUI) as described below. Transmittal of this information on sign-in is appropriate because group information will not change for the duration of the session. When the user signs in, he will receive group information that will be used for the remainder of the session.

Some means of indicating to the user what groups are available and allowing him to select desired recipient groups is required as well. A GUI was designed for this purpose. Previous implementations of the TwiddleNet Client software already had a form-based System Setup GUI to allow the user to set system configuration parameters. Ableiter covers this GUI, including screenshots [25]. For this design a new tab was added to this form, labeled "RecipientOpt," short for "Recipient Options." Figure 4 shows what this GUI would look like on a device running Windows Mobile 5.

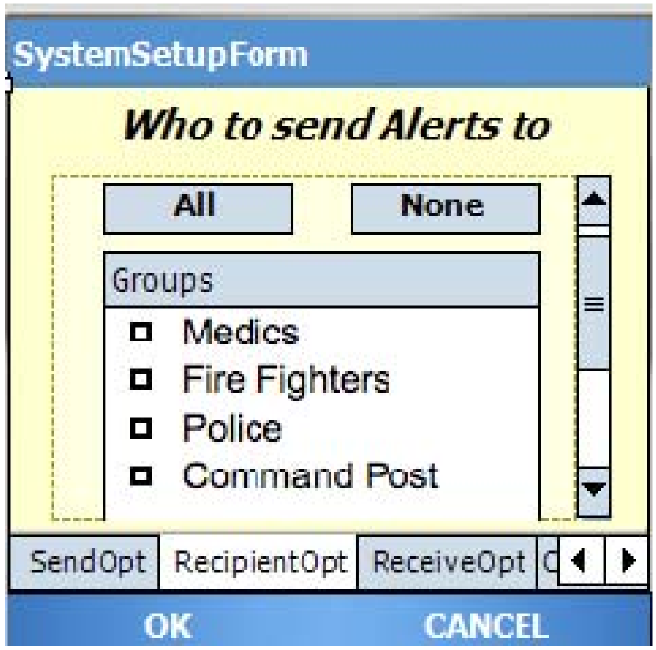


Figure 4. TwiddleNet Client Recipient Options tab

The new tab displays the available TwiddleNet groups to the user and allows him to specify the groups that will receive a new alert by selecting a checkbox. By default, the group the user belongs to will be checked. The list of selected groups is saved internally by the Client software along with the information from other tabs when the user presses the "OK" button. Two other buttons labeled "All" and "None" were added for convenience. These buttons can be used to select or deselect all of the groups on the list at once. The combination of groups selected by the user determines how alerts are generated by the Portal, as detailed in Chapter IV. For consistency sake, the look and feel of the tab, including object colors, font style, and font size were selected so as to match the style used on the other tabs in the form.

After the user configures the Recipient Group options, this information is included in future notifications of new content that the Client sends to the Portal. This indicates to the Portal the groups that need to be alerted for the new content. The Portal will use this information to generate its alert address list as discussed below.

2. Portal

a. Database Design

The TwiddleNet Portal's database is absolutely essential for the operation of the system as a whole. The database is used to store important system information including user credentials, group membership information, data about the devices being used in the system, such as IP address, and metadata for content that has been shared. The work for this implementation included a redesign of the database, preserving existing functionality while adding the functionality necessary to support user groups. Formal database design methods were used in the creation of this database, including Entity-Relationship (E-R) and Relational modeling. Details of the design are presented in the Appendix, including the E-R diagram, Relational model, and the Data Dictionary describing the database's tables and attributes in detail. Important aspects of the database are highlighted below.

(1) Portal Users Table. This table contains user identification information necessary for verification during the user sign-in process as well as linking users to groups in order to track group membership. The Portal Users Table, along with the Belongs To and Group Tables, is fundamental to the user-partitioning feature of this implementation.

(2) Belongs To Table. This table maps users to groups by linking a unique user identifier from the Portal Users Table with a unique group identifier in the Groups Table. This allows the Portal to know what users belong in each group so an alert can be properly addressed when it is destined for a certain group.

(3) Groups Table. This table stores the various groups available in the system.

(4) Uses Table. This table links a particular user with the device he is using. This is done by linking a unique identifier for the user from the Portal Users

Table with a device identifier in the Devices Table. When an alert needs to go to a specific user, the IP address of the device being used by that user can be retrieved from the Devices Table via this linkage.

(5) Devices Table. This table stores important identification information, including IP address, for the mobile devices being used in the TwiddleNet system.

(6) Content Info Table. This table stores the metadata associated with content that has been shared.

(7) Special Tags Table. This table is meant to store situation-specific information as determined by the system administrators and mission planners using the system. The current TwiddleNet implementation makes use of this table to store triage information as detailed in [25].

The database management system used for this implementation was MySQL. This choice was driven by the fact that MySQL is open-source and freely available. Moreover, MySQL allows for easy administration through the use of tools such as phpMyAdmin [31].

b. Sign-in Process

As mentioned above, the design of the sign-in process was modified as part of this implementation. After verifying the user's credentials, instead of simply sending an acknowledgement, the Portal now performs a database lookup and then sends a list of all available user groups, as well as the group to which that particular user belongs, to back to the Client. Details of this process are included in Chapter IV.

c. Alert Addressing Process

In previous TwiddleNet implementations the Portal would simply send an alert to all signed-in users when notified of new content being available. This required the Portal to generate a list of the IP addresses for all active client devices. The Portal did this by retrieving the IP addresses from its database. In this design the process was

modified so as to send an alert only to users in the groups indicated by the user sharing the content. The Portal now creates its alert recipient address list by gathering the IP addresses for users that are signed-in and who belong to one of the groups indicated in the new content notification. This process is discussed in detail in Chapter IV.

3. Command Post

No major changes to the Command Post software were necessary for this implementation. Administratively, the Command Post was placed in its own group in order to provide users a means of sharing content directly with it. The Command Post won’t receive an alert unless the user selects the Command Post group in the GUI as discussed earlier.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. IMPLEMENTATION AND TESTING

A. IMPLEMENTATION TOOLS

1. Software

As this implementation builds upon past implementations which were written in the C# programming language, the programming for this work was also done in C#. The TwiddleNet Portal code is a console application based on the .NET 2.0 Framework, designed to be run on a PC. The TwiddleNet Client code is a Pocket PC application based on the .NET 2.0 Compact Framework, designed to be run on devices using the Windows Mobile 5 operating system.

Microsoft's Visual Studio 2005 Integrated Development Environment was used for development of the TwiddleNet software. Visual Studio is an excellent tool for development using .NET-based languages like C#. When integrated with the Pocket PC Software Development Kit, Visual Studio becomes a powerful platform for mobile development with built-in tools for easy deployment of code to the device and emulators for testing without an actual device.

The MySQL Database Management System was used to host the TwiddleNet Portal's database. In this implementation, MySQL was provided as part of the XAMPP [32] software suite. XAMPP is a cross-platform Apache-MySQL-PHP-Perl implementation that is very easy to setup and use. Using XAMPP, one can setup a web and database server simply by downloading the software and unpacking it. The software is self-contained and preconfigured so that the Apache web server supports PHP, allowing tools like phpMyAdmin to be used out of the box. Use of this software suite allows a TwiddleNet Portal to be setup quickly and easily.

A few additional software tools are useful for the creation and management of the TwiddleNet Portal's database. As previously mentioned, the Apache web server that is

part of the XAMPP suite allows for the use of phpMyAdmin, providing a web-based interface useful for the administration of MySQL databases. A new database can easily be created using this tool.

Once an empty database for the Portal has been created, its structure must be defined and certain essential fields must be populated. Two SQL scripts were created for this purpose (see the Appendix). These scripts contain the SQL commands necessary to create the database structure and perform the initial population. Once these scripts have been run, the Portal database is ready for use.

Taking advantage of the Apache web server included in the XAMPP package, a web-based interface, written in PHP, was created to simplify the management of the TwiddleNet Portal database. This administration web page can be used to create new TwiddleNet users and groups, assign users to groups, add devices to the database, and assign users to devices. Examples of the interface are seen in Figures 5 and 6. Figure 5 shows what the User Listing page looks like and Figure 6 shows the page that is used to add a new user.

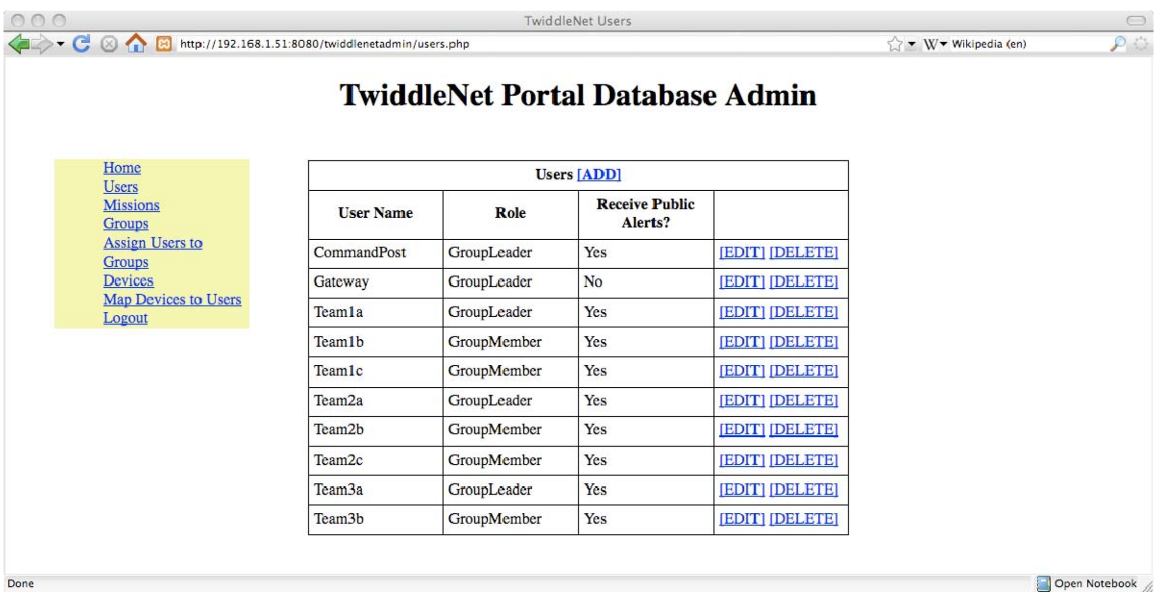


Figure 5. TwiddleNet Portal Database Admin Page, User Listing

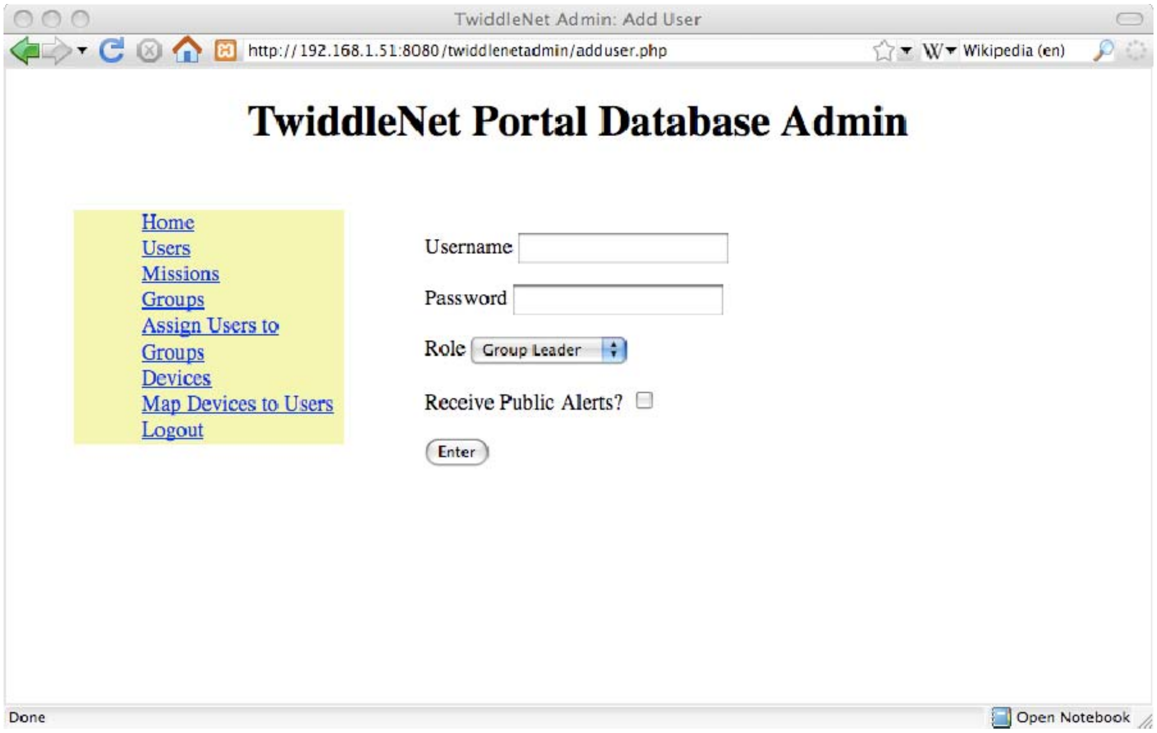


Figure 6. TwiddleNet Portal Database Admin Page, Creation of New User

2. Hardware

The mobile devices used to run the TwiddleNet Client software are HP iPAQ hw6945 smartphones. These devices are quite powerful with a large touch screen display, Qwerty keyboard, and multiple networking interfaces including WiFi (802.11 WLAN), GSM (WWAN), and Bluetooth (PAN). The devices use the Windows Mobile 5 operation system and run software based on the .NET Compact Framework.

The TwiddleNet Portal can be run on any Windows machine capable of supporting the .Net 2.0 Framework. During development and testing this was typically a Windows XP laptop or an OQO Ultra Mobile PC also running Windows XP. The small size and portability of the OQO makes it convenient for use in a mobile system like TwiddleNet.

For demonstration and testing the TwiddleNet machines are provided IP addresses via the Dynamic Host Configuration Protocol (DHCP). A laptop running Windows

Server 2003 is used for this purpose. For convenience, the Command Post software is also run on this machine along with the Apache web server that supports it.

In the lab an 802.11 access point is typically used to create a stand-alone WiFi network for demonstration and testing. In actual usage an access point in a host network or some form of mobile access point may be utilized.

3. Lab Network

Figure 7 shows how the TwiddleNet system is arranged in the lab for testing and demonstration purposes. The access point creates a stand-alone WiFi network. The DHCP server provides IP addresses to the various components. The Access Point, Portal, and Command Post use reserved addresses while the Clients receive address dynamically from a specific range.

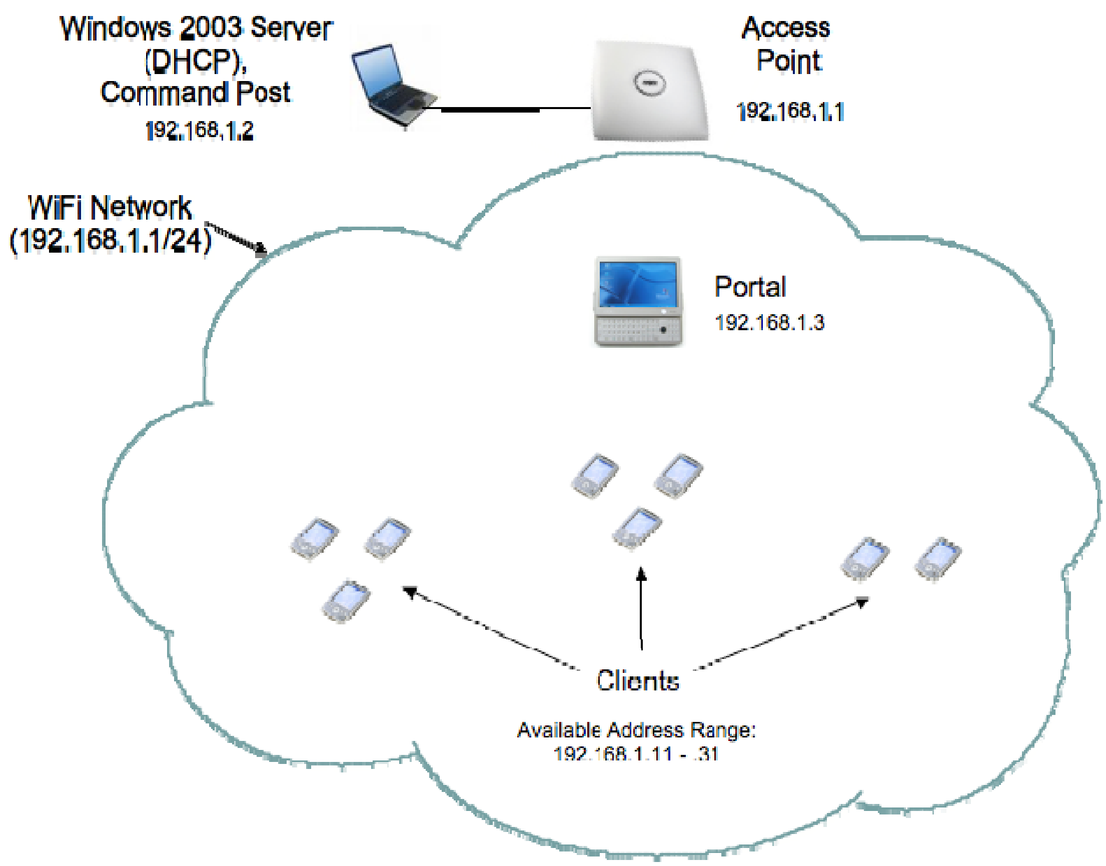


Figure 7. TwiddleNet Lab Network

B. NEW TWIDDLENET SYSTEM OPERATION

1. Overview

The following is a high-level overview of the new TwiddleNet System operation. Figure 8 illustrates the user-partitioning feature of the new implementation. Here the Client devices are organized into three separate groups instead of a single group as in Figure 1.

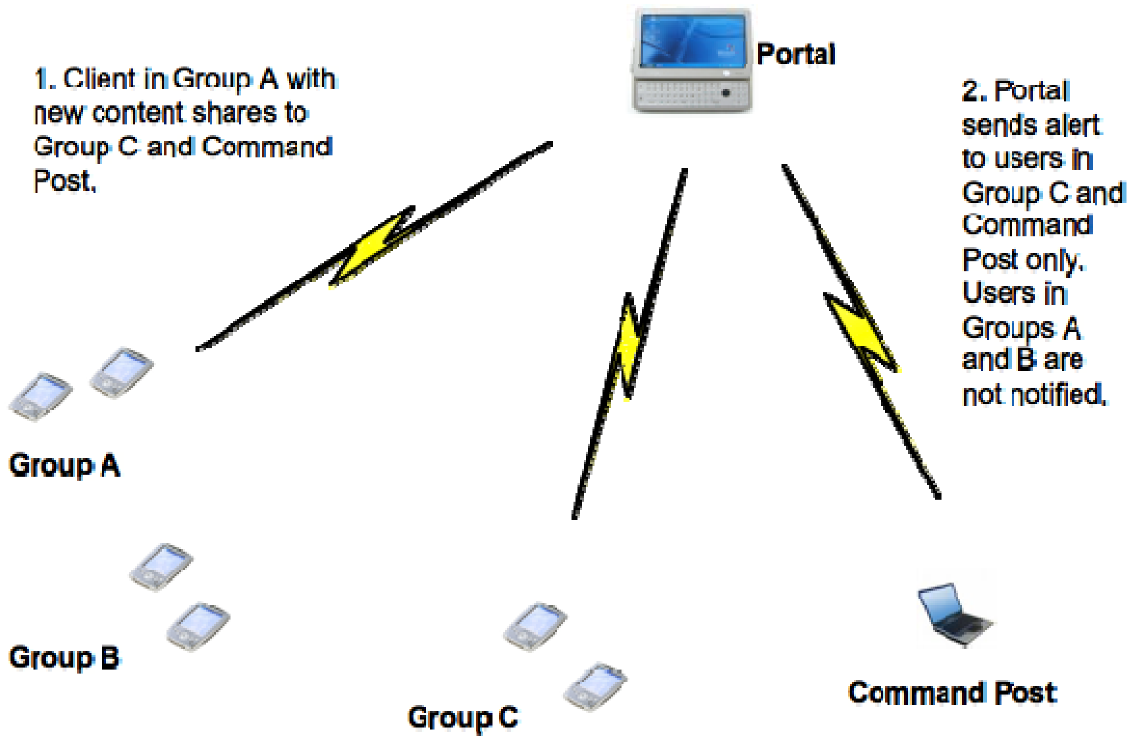


Figure 8. New TwiddleNet System Operation

The new system operation can be illustrated using an information sharing scenario like that discussed in Chapter II. Here, unlike in the previous discussion, the TwiddleNet users are broken up into separate groups, named A, B, and C, for the purpose of discussion. The Command Post is assigned to a separate group on its own. In this scenario a user in Group A wishes to share content with users in Group C and the Command Post but not with other users in Group A or with those in Group B.

The scenario begins with the user starting the TwiddleNet Client software and signing into the Portal. By the time the sign-in process is complete, the Client will have received the group information necessary to populate the Recipient Options tab of the System Setup GUI (discussed in Chapter III) from the Portal. The user would then select Group C and the Command Post group while leaving Groups A and B unselected. Now the Client software is configured according to the user's preferences for alert recipients.

When the user shares content, like in previous implementations, a notification is sent to the Portal containing metadata for the shared content. In the new implementation this metadata also contains the groups specified by the user for alert receipt, namely Group A and the Command Post group (Figure 8, Step 1).

After the Portal receives the notification of new content, it sends an alert to the Clients in the specified groups only, instead of all other active Clients. So Clients in Group C and the Command Post are notified that new content is available (Figure 8, Step 2). Other users in Group A as well as those in Group B will not receive an alert and will therefore not have access to that content. After receiving an alert, a user wishing to download the content may do so as discussed in Chapter II.

2. System Operation Description

This section contains a more detailed description of important aspects of the new TwiddleNet System operation. For the purposes of this discussion the information-sharing process is broken down into the four major steps: 1) sign-in, 2) Recipient Options configuration, 3) notification of new content, and 4) alert generation and transmittal.

a. Sign-in

The sign-in process (illustrated in Figure 9) begins with the user entering his user name and password when prompted by the TwiddleNet Client software running on the handheld device. The Client then creates a TCP connection with the Portal and sends the user name and password entered by the user, the device's MAC addresses, and two hash values, for the user name and password, respectively. The hashed user name is used as a unique identifier for Portal Users records in the Portal's database (see

Appendix). As a security precaution, the Portal stores a hash value of the password to avoid keeping a plain-text version. The MAC address is currently not used, but was considered by previous TwiddleNet developers to be potentially useful for future software features.

After receiving this information the Portal processes it. The first step in processing is to validate the user by comparing the user name and password hashes it was passed to those stored in its database. If the passed-in values don't match, the Portal will send an error message to the Client. The Client will then prompt the user to try the sign-in again or quit.

If the user is successfully validated, the Portal will then check the device IP address stored in the Devices record (see Appendix) associated with the user and update it if necessary (the Portal has access to the device's current IP address via the TCP connection). This is one way in which the Portal keeps track of the current IP address of active devices. For further details on Client IP address tracking see [25].

Next, the Portal will retrieve the name of the group the user belongs to as well as a list of all existing groups. This information is then sent to the Client along with an acknowledgement message to indicate successful sign-in. The Client will store this group information internally and use it to build the Recipient Options tab on the System Setup GUI as discussed in Chapter III.

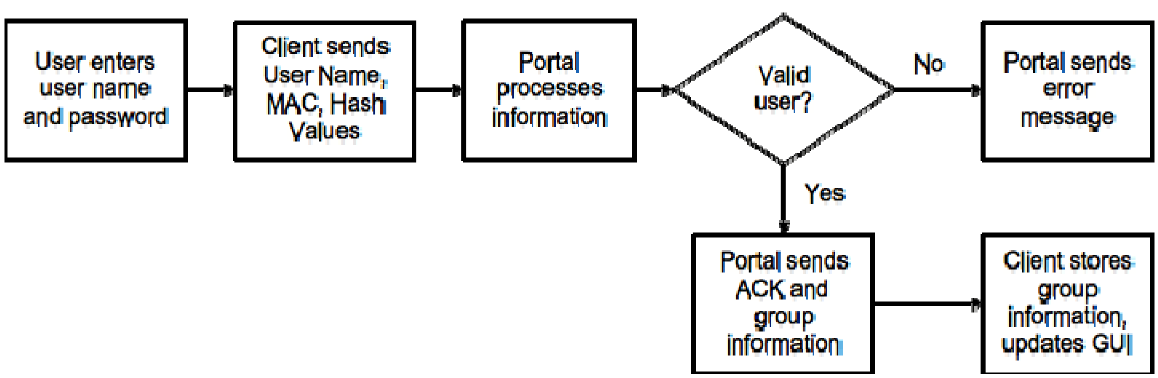


Figure 9. User Sign-in Process

b. Recipient Options Configuration

As previously mentioned, the TwiddleNet Client software uses the group information it receives from the Portal as part of the sign-in process to build the Recipient Options GUI (see Figure 4) that the user will use to indicate which groups should receive alerts for future shared content. By default, only the user's group will be selected, so if he makes no configuration changes, content will be shared only with the other users in his group. If the user desires to share with groups other than his own, he will call up the Recipient Options configuration and select additional groups. If all groups are selected, a special flag is set indicating the content is public. After the user presses the "OK" button, the selected group names will be saved internally for future use by the Client software when building notifications destined for the Portal.

c. Notification of New Content

When a user shares content, the TwiddleNet Client software builds a notification message for that item. This message is in XML format and includes metadata describing the content as well as the recipient group information saved earlier. An example of a notification message is shown below.


```
<?xml version="1.0" encoding="utf-8"?>
<TNetMessage>
  <content>
    <user_name>Team1a</user_name>
    <task>SHARE</task>
    <tags>
      <filename>T-Ten0016.jpg</filename>
      <file_size>171805</file_size>
      <file_type>jpg</file_type>
      <public>True</public>
      <date_created>2008-09-24T16:49:24</date_created>
      <date_shared>2009-01-15T12:09:54</date_shared>
      <rec_groups>
        <group_list>
          <group_name1>CommandPost</group_name1>
          <group_name2>Fire Fighters</group_name2>
          <group_name3>Medics</group_name3>
          <group_name4>Police</group_name4>
        </group_list>
      </rec_groups>
```

```
      <special_tags>
      </special_tags>
    </tags>
  </content>
</TNetMessage>
```

This example shows that a picture (T-Ten0016.jp) has been shared with four groups included in the <group_list> tag. These four groups happen to represent all the groups implemented in the system so the <public> tag contains the value "True." The value of this flag affects the behavior of the Portal when it processes notifications as discussed below.

After the notification message has been built, the Client opens a TCP connection with the Portal and sends the message. The Portal parses and stores the metadata contained in the message, then responds with an acknowledgement or an error message if the notification couldn't be processed properly. When the Client receives an acknowledgement, the GUI is updated indicating the content was successfully shared. Otherwise an error message is displayed to the user. The notification process is illustrated in Figure 10.

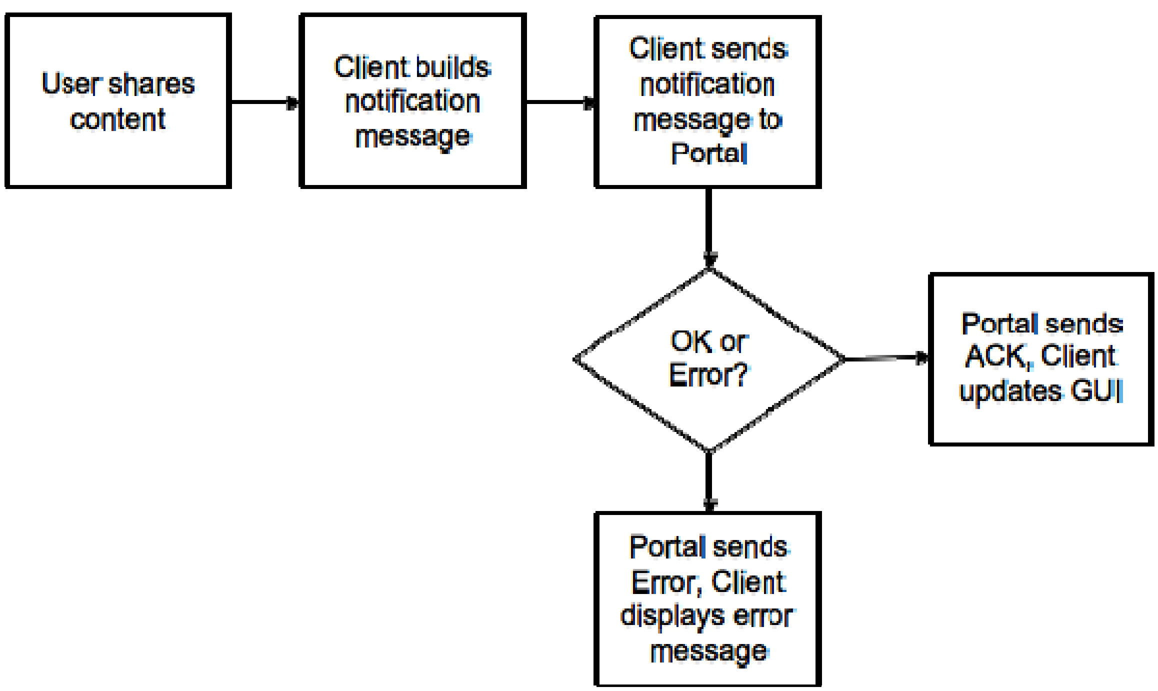


Figure 10. Client-Portal Notification Process

d. Alert Generation and Transmittal

When the Portal receives a notification message for new content, it must process the information. This includes parsing the message and storing the metadata found therein in a new ContentInfo record in its database (see Appendix). The Portal also stores the recipient groups contained in the notification message internally for future use in alert generation.

After the incoming information has been processed, the Portal builds the alert message that will be sent to other users alerting them of the availability of new content. The alert message is also an XML message in the same format as the notification message. Unlike in previous implementations where the Portal simply sent an alert to each active user, here the Portal sends the alert only to the users belonging to the groups specified in the notification message. In order to achieve this the Portal uses a nested SQL query to first pull the user names belonging to the groups included in the notification message from its database, and then pull the IP addresses associated with the devices assigned to all active users in that list. The result is a list containing the IP address associated with the device being used by each user belonging to the recipient groups indicated by the user sharing the content.

After the IP address list has been generated, the Portal sends the alert message to each user's Client via the address for their device on the list. As in previous implementations, this is done in a multithreaded manner freeing the main Portal process so that it may handle other communications and ensuring that alerts are sent in a timely manner.

C. IMPLEMENTATION ISSUES

In this section notable implementation-specific issues are discussed. For this implementation a few changes were necessary to address issues and make improvements in the TwiddleNet code. Theses changes involved the use of hash functions as well as XML parsing methods.

1. Hash Functions

In the previous implementation a .NET-specific hash function was used to generate hash values for user names and passwords. This did not pose a problem in the past because the only software that handled these values was based on the .NET libraries. The new implementation, however, allows for user records of the Portal database, including user name and password hashes, to be created using the previously mentioned web-based administration interface. When user records are created using this tool, hash values must be created using functions available in PHP. Since the algorithm used by the .NET hash function was unknown, and therefore impossible to replicate using PHP, the author was unable to create a hash value in PHP that matched the value produced by the .NET function when hashing the same string. This affected the user sign-in process, as user validation was impossible for users created via the web-based administration tool. Since MD5 hashes can be created in both .NET and PHP, the TwiddleNet code was modified to use an MD5 hash function. In this way a string, such as a user name or password, when hashed by a PHP function, produces the same value as that produced by the .NET hash function when hashing the same string, and can be compared during user validation without problems.

2. XML Parsing

Much of the Client-Portal communication in TwiddleNet relies on the use of XML. Consequently, the Client and Portal code must be able to parse XML documents in order to retrieve the information embedded within. The previous implementation was rather inflexible in the manner in which this parsing was implemented. The code that parsed an XML document depended heavily on the structure of the document. It used nested if-else statements that corresponded directly to the structure of the XML document. In other words, if the XML document had tags that were nested three levels deep, the parsing code had to have if-else statements nested three levels deep. Thus, any change to the structure of the XML document necessitated a great deal of work changing the parsing code.

The issue was addressed in this implementation through the use of a recursive XML parsing method. This makes the parsing code independent of the structure of the XML document itself. The method can parse an XML document no matter how many levels of nesting are used. Furthermore, the code does not need to be rewritten if the structure of the XML document is changed. This will ease the burden on programmers as future changes are made to the system.

D. TESTING

The new TwiddleNet implementation was tested in January 2009 as part of the Cooperative Operations and Applied Science and Technology Studies (COASTS) Field Experiment II (FEX-II) at Camp Roberts, California. TwiddleNet researchers have been involved with the COASTS project for some time as it provides an excellent test bed. The COASTS field experiments give researchers an opportunity to run TwiddleNet in a real-world environment outside the confines of the lab.

Rather than being run in its own self contained network, as it is for lab testing and demonstration, during the FEX TwiddleNet was integrated into the greater Camp Roberts network. Figure 11 shows how the TwiddleNet network was arranged during the FEX testing.

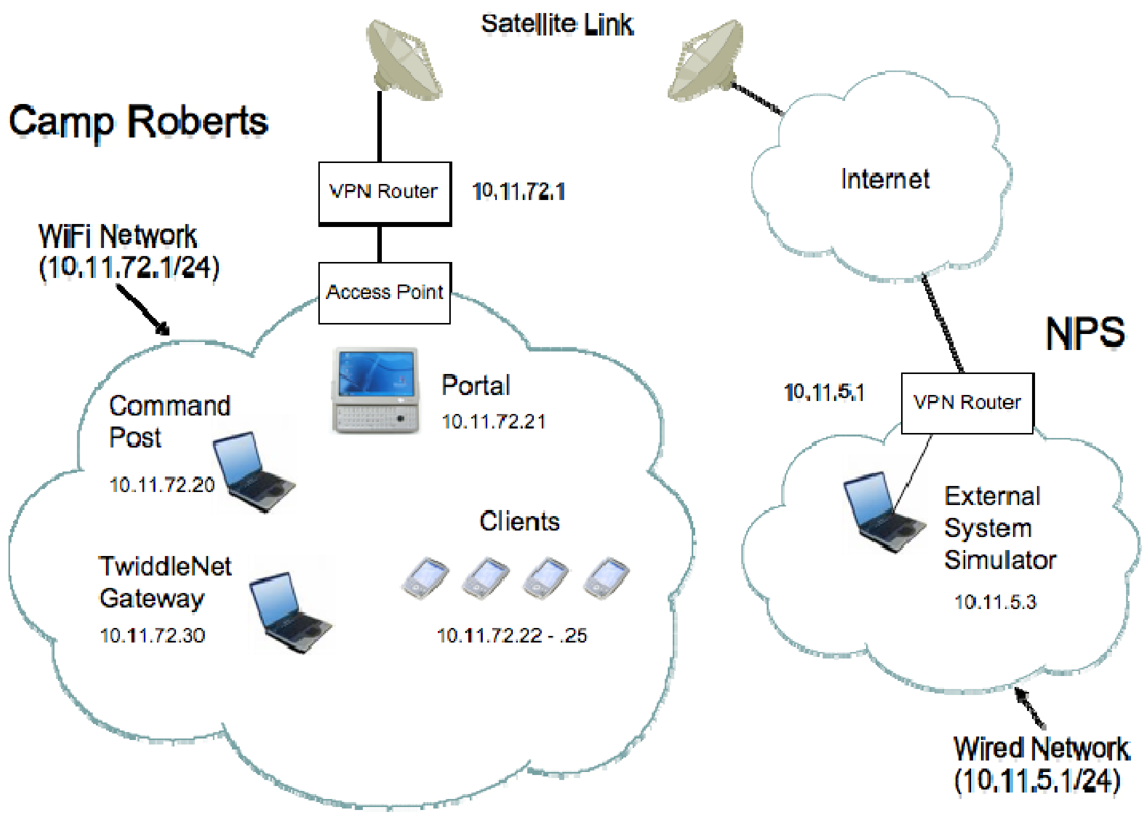


Figure 11. TwiddleNet Integration into Camp Roberts Network

TwiddleNet was assigned a block of addresses belonging to the Camp Roberts network for use during the FEX. All TwiddleNet components were assigned static addresses from this block as seen in Figure 11. An access point was available to provide connectivity via WiFi.

The user-partitioning feature of the new TwiddleNet implementation was successfully tested during the FEX. Users were organized into three different groups and content was shared to various combinations of groups. The only difficulty experienced had to do with the distance of devices from the access point. If the devices were more than approximately 60 yards away, content transfers between the devices and from the devices to the Command Post failed. It is believed that this is due to the relatively lower signal strength at the periphery of the access point's coverage range.

Another notable aspect of the FEX testing centered on a new tool called the TwiddleNet Gateway. The TwiddleNet Gateway is a piece of software that interfaces

TwiddleNet with external systems allowing content to be shared from sources other than TwiddleNet Clients. While not necessarily related to the work done for this thesis on privacy protection, the Gateway software does take advantage of the user-partitioning feature developed as part of this work. This allows the Gateway to share content it receives from external sources with TwiddleNet users on a per-group basis much like a standard TwiddleNet Client can.

The TwiddleNet Gateway was included as part of the TwiddleNet network during the FEX testing (see Figure 11). Content was shared with the Gateway over a VPN connection with the wireless lab at the Naval Postgraduate School (NPS) using the External System Simulator (ESS). The ESS is a simple program designed to simulate a content feed that might be provided from an external system. The ESS provides a notification of new content to the Gateway, which then retrieves the content and shares it with other TwiddleNet users.

Overall the FEX testing was very successful. The ease at which TwiddleNet is able to be integrated into other networks like the Camp Roberts network is due in no small part to the decisions made by previous designers, who chose to base all TwiddleNet communications on universal standards such as TCP/IP and HTTP.

V. CONCLUSION AND FUTURE WORK

A. CONCLUSION

This work successfully implemented privacy protection for TwiddleNet, a network of mobile devices intended for use by first responders. As the user-partitioning feature of this implementation shows, network virtualization is an appropriate choice for implementing privacy in mobile networks.

As discussed in Chapter I, this work focused on the requirement of content privacy. This implementation partially meets this requirement in that content is protected from inadvertent release to unauthorized users, but not protected from eavesdropping. In other words, the threat from casual observers is addressed, but the attacker threat is not addressed. An attacker, with advanced knowledge of networking and the use of specialized “sniffing” tools, could still gain access to content shared by TwiddleNet users, compromising their privacy. Properly addressing this threat calls for a more sophisticated approach, perhaps incorporating techniques mentioned in Chapter II, and is left to future work.

In just a few short years TwiddleNet has become a robust and useful system, improving with each incremental development step in order to better meet the needs of first responders. This work realizes the latest step forward in that progression. It is the sincere hope of the author that this progression continues for years to come.

B. FUTURE WORK

This work represents the first step toward improving the overall security posture of the TwiddleNet System. Much more can be done to improve the privacy, security, and availability of the system. Some ideas for future work are presented below.

1. End-to-End Encryption

Currently, even if mechanisms such as WEP or WPA are enabled in order to protect content in the wireless medium, the system could still be vulnerable to an attacker capturing packets on any wired links. An end-to-end encryption scheme utilizing technology like Secure Sockets Layer (SSL) could be used to address this. Furthermore, encryption keys used in such a scheme could also be used in support of authentication.

The use of SSL proves challenging, however, due to the implementation platform used for the TwiddleNet Client software, as the .NET Compact Framework 2.0 does not support SSL connections. Third-party libraries, such as those provided by UDAParts [33], may provide a way forward in SSL-enabling TwiddleNet communications.

2. Addressing the Single-Point-of-Failure Issue

The TwiddleNet Portal represents a potential single-point-of-failure in the system. If the Portal fails, all content metadata, as well as user and group information, will no longer be accessible. Furthermore, client communications will no longer be possible as the Portal is central to all sharing, alerting, and transfer of content. Providing a mechanism for redundancy or making the system architecture more distributed could greatly improve the overall availability of the system.

3. Software Engineering

The TwiddleNet project could benefit greatly from a formal software engineering effort. This should include requirements definition, use case analysis, and formal UML documentation. This would not only help newcomers to the project who need to learn the software, but also would provide a basis for future implementation on platforms other than .NET. The design should be done with extensibility in mind so that future modifications and additions are easier. In addition, the documentation should be updated as new features are added or other changes are made.

4. Command Post Improvements

A few improvements to the Command Post software would go a long way toward increasing its usefulness. Currently the web page that the software builds is mostly static. There is the ability to append information to the metadata displayed, but no new content is shown until the page is refreshed. Implementing the page using technology like Asynchronous Java and XML (AJAX) could address this. With AJAX, a portion of a web page can be updated without refreshing the entire page.

The current implementation of the Command Post is receive-only in that it displays only content shared from TwiddleNet clients but cannot share content itself. Another potential improvement to the software would be to add the ability to share content just like a standard TwiddleNet Client. A technology like AJAX could possibly be used here as well, creating a web-based application that could be used to view content and its associated metadata, update that information, and send the updates back out to Clients.

5. Integration with Other COASTS Systems

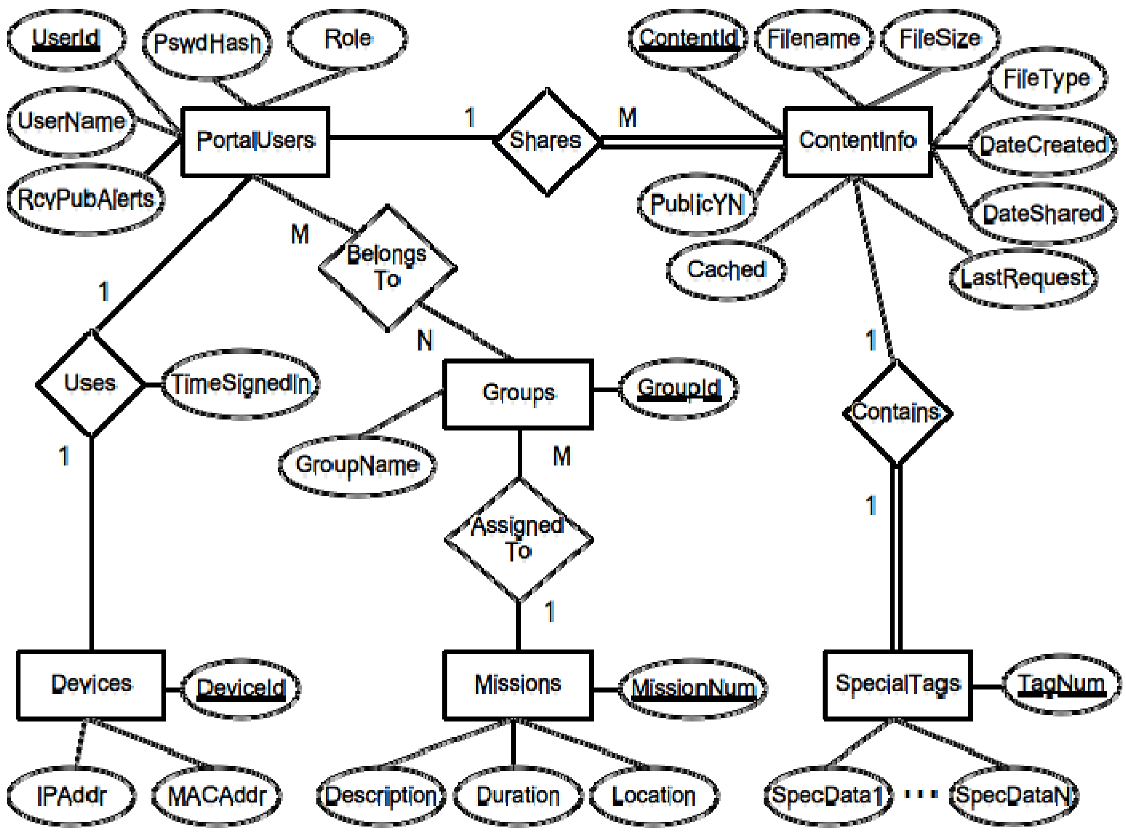
There may be great benefit derived from the integration of TwiddleNet with other systems that are part of the COASTS program, specifically the Real-Time Automated Position Identification System (RAPIDS) and the perimeter security system being tested during COASTS experiments.

RAPIDS is a three dimensional positioning system used for situational awareness and command and control purposes. Using RAPIDS, the position of an Unmanned Aerial Vehicle (UAV), for example, can be tracked on a common operational picture. Integrating TwiddleNet with RAPIDS could allow images taken from the UAV to be distributed to teams on the ground, potentially providing real-time intelligence or other tactical data.

TwiddleNet could also be integrated with the perimeter security system used in COASTS experiments. For example, TwiddleNet could be used to distribute pictures of contacts of interest, such as suspicious vehicles or illegal immigrants, to border security teams.

APPENDIX

TwiddleNet Portal Database Entity-Relationship Diagram



TwiddleNet Portal Database Relational Model

PortalUsers (<u>UserId</u>, UserName, PswdHash, Role, RcvPubAlerts)

Groups (<u>GroupId</u>, GroupName, *MissionNum*)

BelongsTo (<u>*UserId, GroupId*</u>)

Devices (<u>DeviceId</u>, IPAddr, MACAddr)

Uses (<u>*UserId*</u>, *DeviceId*, TimeSignedIn)

Missions (<u>MissionNum</u>, Description, Duration, Location)

ContentInfo (<u>ContentId</u>, Filename, FileSize, FileType, Public, Cached, DateCreated, DateShared, LastRequest, *TagNum*, *UserId*)

SpecialTags (<u>TagNum</u>, SpecData1, … , SpecDataN)

TwiddleNet Portal Database Data Dictionary

Assumptions

- A user must belong to at least one group.
- UserIds are unique.
- A user may be sharing zero or more pieces of content at any given time.
- A particular file may be shared by more than one user, but one ContentInfo record is associated with only one user.
- A user can't share the same content more than once.
- A user can only log in to the system from one device at a time.
- A device that isn't being used to run the TwiddleNet application isn't associated with the system.
- A device can only have one IP address at a given time (if both WiFi and GPRS radios are enabled at the same time, the device will associate with the WiFi access point only).
- A group can only be assigned to one mission at a time.

Data Dictionary

Entity: PortalUsers – All the users currently registered to use TwiddleNet
Attributes:

- UserId – A unique identifier for the user signed into TwiddleNet.
- UserName – The name of the user identified by UserId.
- PswdHash – The value resulting from hashing the user's password.
- Role – The role the particular user plays within the group.
- RcvPubAlerts – Boolean value indicating whether or not this user should receive public alerts.

Entity: Groups – The different groups the users are organized into.
Attributes:

- GroupId – A unique identifier assigned to the group.

- GroupName – The name of the group. Ex: "Red Cross".

Entity: Devices – The devices currently associated with the system.
Attributes:

- DeviceID – A unique identifier for the device; used for database implementation purposes.
- MACAddr – The MAC address of the device being used by the associated user.
- IPAddr – The IP address currently assigned to the device being used by the associated user.

Entity: ContentInfo – Information describing the content that the associated user is sharing.
Attributes:

- ContentId – A unique identifier for the content.
- Filename – The filename of the content being shared.
- FileSize – The size (in memory) of the file.
- FilesType – The file extension of the content.
- Public – A boolean indicating whether or not the content should be available to all users or just the group associated with the sharer.
- Cached – A boolean indicating whether or not the content is cached on the Portal.
- DateCreated – Date the content was created.
- DateShared – Date the content was shared.
- LastRequest – Date and time the content was last requested from the sharer.

Entity: SpecialTags – Extended information describing the content specific to a particular situation.
Attributes:

- TagNum – A unique integer identifying a particular SpecialTags record.
- Only generic attributes are named at this time. For example SecData1, etc. for Special Data Items.

Entity: Missions
Attributes:

- MissionNum – A unique identifier for the mission.
- Description – A description of the mission. For instance, purpose.
- Duration – Number of hours the mission is expected to last.
- Location – The geographic location the mission is to take place at.

Relationships:

- BelongsTo – Relates users to the groups they belong to.
- Uses – Relates a user to the device that user is currently using.
    - Attributes:
        - TimeSignedIn – The time the user signed into TwiddleNet.

- Shares – Relates particular ContentInfo items to the user sharing the associated content.
- Contains – Relates SpecialTags with the associated ContentInfo.
- AssignedTo – Relates missions with groups that are performing them.

Portal Database SQL Scripts

Script to create empty database:

```
CREATE TABLE portalusers (
	user_id CHAR(32) PRIMARY KEY,
	user_name VARCHAR(100) UNIQUE,
	pswd_hash CHAR(32),
	role ENUM ("GroupLeader", "GroupMember"),
	rcv_pub_alerts BOOLEAN
	);

CREATE TABLE missions (
	mission_num VARCHAR(10) PRIMARY KEY,
	description TEXT,
	Duration INT,
	Location VARCHAR(100)
	);

CREATE TABLE groups (
	group_id INT PRIMARY KEY AUTO_INCREMENT,
	group_name VARCHAR(100),
	mission_num VARCHAR(10),
	CONSTRAINT groups_msn_num_fk FOREIGN KEY (mission_num)
		REFERENCES missions(mission_num) ON DELETE SET NULL
		ON UPDATE CASCADE
	);

CREATE TABLE belongsto (
	user_id CHAR(32) NOT NULL,
	group_id INT,
	CONSTRAINT bel_to_user_id_grp_id_pk PRIMARY KEY (user_id, group_id),
	CONSTRAINT bel_to_user_id_fk FOREIGN KEY (user_id)
		REFERENCES portalusers (user_id) ON DELETE CASCADE
		ON UPDATE CASCADE,
	CONSTRAINT bel_to_grp_id_fk FOREIGN KEY (group_id)
		REFERENCES groups (group_id) ON DELETE CASCADE
		ON UPDATE CASCADE
	);
```

```
CREATE TABLE devices (
	device_id INT PRIMARY KEY AUTO_INCREMENT,
	ip_addr VARCHAR(100) DEFAULT 'TempValue',
	mac_addr VARCHAR(100) DEFAULT ' TempValue',
	batt_level INT DEFAULT 0,
	device_sn VARCHAR(30) UNIQUE DEFAULT ' TempValue'
	);

CREATE TABLE uses (
	user_id CHAR(32) PRIMARY KEY,
	device_id INT,
	time_sig_in TIMESTAMP,
	CONSTRAINT uses_user_id_fk FOREIGN KEY (user_id)
		REFERENCES portalusers(user_id) ON DELETE CASCADE
		ON UPDATE CASCADE,
	CONSTRAINT uses_device_id_fk FOREIGN KEY (device_id)
		REFERENCES devices(device_id) ON DELETE NO ACTION
		ON UPDATE CASCADE
	);

CREATE TABLE specialtags(
	tag_num INT PRIMARY KEY AUTO_INCREMENT,
	gender VARCHAR(100),
	age VARCHAR(100),
	last_name VARCHAR(100),
	first_name VARCHAR(100)
	);

CREATE TABLE contentinfo(
	content_id INT PRIMARY KEY AUTO_INCREMENT,
	filename VARCHAR(100),
	file_size VARCHAR(100),
	file_type VARCHAR(10),
	public_yn BOOLEAN,
	cached BOOLEAN DEFAULT false,
	date_created DATETIME,
	date_shared DATETIME,
	last_request DATETIME,
	tag_num INT NOT NULL,
	user_id CHAR(32) NOT NULL,
	CONSTRAINT contentinfo_tag_num_fk FOREIGN KEY (tag_num)
		REFERENCES specialtags(tag_num) ON DELETE CASCADE
		ON UPDATE CASCADE,
	CONSTRAINT contentinfo_user_id_fk FOREIGN KEY (user_id)
```

```
        REFERENCES portalusers(user_id) ON DELETE CASCADE
        ON UPDATE CASCADE
    );
```

Script to create initial database information:

```
INSERT
INTO specialtags (gender, age, last_name, first_name)
VALUES ('None supplied', 'None supplied', 'None supplied', 'None supplied')
```

LIST OF REFERENCES

[1] T. Virki, “Global cell phone use at 50 percent,” November 29, 2007, http://www.reuters.com/article/technologyNews/idUSL2917209520071129, last accessed November 2008.

[2] “Europeans hang up on fixed lines,” November 28, 2007, http://news.bbc.co.uk/2/hi/technology/7116599.stm, last accessed November 2008.

[3] N. Ai, Y. Lu and J. Deogun, "The smart phones of tomorrow," *SIGBED Rev.*, vol. 5, pp. 1-2, 2008.

[4] S. Ferguson and C. Boulton, “The future is mobile,” January 14, 2008, http://www.eweek-digital.com/eweek/20080114_sec/?pg=12, last accessed November 2008.

[5] D. Wisley, “Getting the Right Information to the Right Place at the Right Time,” May 15, 2002, General Dynamics Decision Systems, http://www.niusrjournal.org/n_core/pdf/Effective_interoperability.pdf, last accessed January 2009.

[6] Report of the Committee on Privacy and Related Matters, Chairman David Calcutt QC, 1990, Cmnd. 1102, London: HMSO, p. 7.

[7] H. van Kranenburg, “Privacy aspects in Internet and mobile services,” November 28, 2000, https://doc.freeband.nl/dsweb/Get/Document-13064/D1.1.6%20privacy_SOTA.doc, last accessed October 2008.

[8] T. Aura and A. Zugenmaier, “Privacy, control and internet mobility,” in *Security Protocols, 12th International Workshop*, April 2004.

[9] B. Askwith, M. Merabti, Q. Shi and K. Whiteley, "Achieving user privacy in mobile networks," in *ACSAC '97: Proceedings of the 13th Annual Computer Security Applications Conference,* 1997, pp. 108.

[10] J. Lindqvist and L. Takkinen, "Privacy management for secure mobility," in *WPES '06: Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, 2006, pp. 63-66.

[11] Health Insurance Portability and Accountability Act of 1996, Public Law 104-191.

[12] Privacy Act of 1974, Public Law 93-579, 5 U.S.C. § 552a.

[13] 5 U.S.C. § 552a(a)(4)

[14] IEEE 802.11i-2004 amendment to IEEE std 802.11, 2004, standards.ieee.org/getieee802/download/802.11i-2004.pdf, last accessed November 2008.

[15] Y. Bin and I. Chlamtac, *Wireless and Mobile Network Architectures*. New York: John Wiley & Sons, 2001.

[16] Electronic Frontier Foundation, "CALEA: The Perils of Wiretapping the Internet," http://www.eff.org/issues/calea, last accessed November 2008.

[17] Electronic Frontier Foundation, "Cell Tracking," http://www.eff.org/issues/cell-tracking, last accessed November 2008.

[18] Electronic Communications Privacy Act of 1986, Public Law 99-508.

[19] Foreign Intelligence Surveillance Act of 1978, Public Law 95-511.

[20] Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism Act, Public Law 107-56.

[21] Communications Assistance for Law Enforcement Act, Public Law 103-414.

[22] G. Singh, "Hastily Formed Networks for First Responders," presented at *13th ICCRTS: C2 for Complex Endeavors*, Bellevue, WA, 2008.

[23] OQO homepage, http://www.oqo.com, last accessed February 2009.

[24] C. Clotfelter and J. Towle, "TwiddleNet: Metadata Tagging and Data Dissemination in Mobile Device Networks," M.S. thesis, Naval Postgraduate School, Monterey, CA, 2007.

[25] D. Ableiter, "Smart Caching for Efficient Information Sharing in Distributed Information Systems," M.S. thesis, Naval Postgraduate School, Monterey, CA, 2007.

[26] R. Moskowitz and P. Nikander. RFC 4423: Host Identity Protocol (HIP) Architecture, May 2006.

[27] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis," *Mob.Netw.Appl.*, vol. 10, pp. 315-325, 2005.

[28] T. Jiang, H. J. Wang and Y. Hu, "Preserving location privacy in wireless lans," in *MobiSys '07: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services,* 2007, pp. 246-257.

[29] J. Lindqvist and J. Tapio, “Protecting Privacy with Protocol Stack Virtualization” to appear in Workshop on Privacy in the Electronic Society (WPES) 2008, October 27, 2008.

[30] S. Cabuk, C. I. Dalton, H. Ramasamy and M. Schunter, "Towards automated provisioning of secure virtualized networks," in *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, 2007, pp. 235-245.

[31] phpMyAdmin homepage, http://www.phpmyadmin.net/home_page/index.php, last accessed January 2009.

[32] XAMPP project homepage, http://www.apachefriends.org/en/xampp.html, last accessed January 2009.

[33] UDA Parts documentation, http://www.udaparts.com/document/articles/demome.htm, last accessed February, 2009.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California